

Efficient Matrix Multiplication Methods to Implement a Near-Optimum Channel Shortening Method for Discrete Multitone Transceivers

Jeffrey Wu, Güner Arslan, and Brian L. Evans

Embedded Signal Processing Laboratory

The University of Texas, Austin, TX 78712-1084 USA

jeffwu78@stanford.edu, arslan@ece.utexas.edu, bevans@ece.utexas.edu

Abstract

The minimum intersymbol interference (min-ISI) method yields time-domain equalizer (TEQ) designs for discrete multitone (DMT) modulation transceivers that are close to channel capacity. For eight standard ADSL channels, the min-ISI design method reaches within 1% of the matched filter bound at the TEQ output. However, the min-ISI method relies several computationally expensive matrix multiplications. In this paper, we develop low-complexity algorithms for these multiplications to allow for real-time implementation of the min-ISI method on programmable digital signal processors.

1. Introduction

Discrete multitone (DMT) modulation is widely used in ADSL and VDSL transceivers, and in digital audio and video broadcasting transceivers, because of their ability to transmit efficiently over spectrally shaped channels. It achieves this goal by dividing the channel into many equally spaced frequency bands, called subchannels, and uses most of these subchannels to transmit data. In this way, the coding can be tailored for each subchannel so that most of the channel bandwidth is used without sacrificing a low bit error rate.

DMT modulation is implemented by the fast Fourier transform (FFT), which naturally leads to processing data in blocks, or symbols. In spectrally shaped channels, one symbol will overlap significantly with the next symbol, causing intersymbol interference (ISI). ISI corrupts the FFT spectrum of the received symbols and thus is a major cause of bit errors in a DMT transceiver.

The first line of defense against ISI is to prepend

the last ν samples of a symbol to the beginning of the symbol. This addition, known as a cyclic prefix (CP), effectively adds a buffer between adjacent symbols while keeping the FFT spectra of the symbols intact. However, the CP must be short because longer CPs decrease the efficiency of a DMT transceiver.

The second line of defense is to add a finite impulse response (FIR) filter, known as a time-domain equalizer (TEQ), after the A/D converter in the receiver. The TEQ attempts to shorten the channel so that the combined impulse response of the channel and TEQ is no longer than $\nu + 1$ sample periods. In this way, any symbol passing through the channel and TEQ will only overlap into the next symbol's cyclic prefix, preventing ISI.

Since it is generally impossible for a TEQ to perfectly shorten the channel to $\nu + 1$ sample periods, one approach to TEQ design is to shorten the channel in such a way that maximizes channel capacity at the TEQ output, e.g. by the Minimum-ISI method [2]. However, the Minimum-ISI method requires computationally complex matrix multiplications. Therefore, the purpose of this paper is to reduce the complexity of these multiplications to allow for real-time implementation of the minimum ISI method on a programmable digital signal processor (DSP).

2. The Minimum-ISI Method

The minimum-ISI method models the channel as an FIR filter with discrete-time impulse response h_k . The estimation of the channel is usually performed in the frequency domain, so the impulse response length is the same as the FFT length in the DMT system, which we denote as N . We represent the additive noise in the channel model with a discrete-time random process n_k . If we let x_k be a random process representing the

transmitted signal, the signal arriving at the receiver is

$$h_k * x_k + n_k$$

where $*$ denotes linear convolution.

The receiver then processes the received signal with the TEQ, which is an FIR filter with an impulse response of w_k . The length of the filter, denoted as N_w , is usually much smaller than N . So at the output of the TEQ,

$$y_k = h_k * w_k * x_k + w_k * n_k$$

The idea behind the minimum-ISI method is to decompose the received signal into its signal, interference, and noise components. Notice that transmitted signal x_k consists of symbols of N points separated by cyclic prefixes of a fixed length ν . Therefore, all of the ISI can be removed if the equalizer can successfully shorten the channel impulse response to $\nu + 1$ samples. With this insight, we formulate the following windowing function g_k to isolate the desired part of h_k

$$g_k = \begin{cases} 1, & \Delta \leq k \leq \Delta + \nu \\ 0, & \text{otherwise} \end{cases}$$

where $\Delta \in \{0, 1, \dots, N - \nu\}$. A fast heuristic method [4] exists to determine *a priori* an estimate of the optimal value of Δ . With this windowing function, we can now separate the signal, interference, and noise components of the received signal as

$$y_k = h_k^{signal} * x_k + h_k^{ISI} * x_k + w_k * n_k \quad (1)$$

where $h_k^{signal} = g_k(h_k * w_k)$ and $h_k^{ISI} = (1 - g_k)(h_k * w_k)$.

The decomposition in (1) enables the following definition of the SNR in each subchannel:

$$\text{SNR}_i = \frac{|H_i^{signal}|^2 S_{x,i}}{|H_i^{ISI}|^2 S_{x,i} + |W_i|^2 S_{n,i}} \quad (2)$$

Note that this definition assumes that both x_k and n_k are wide sense stationary, where S_x is the N -point power spectrum of x_k (without the cyclic prefix) and S_n is the N -point power spectrum of n_k . Since h_k^{ISI} can have up to $N + N_w - 1$ nonzero samples, Thus H^{ISI} is actually the spectrum of the *first* N samples of h_l^{ISI} . However, this small truncation is a reasonable approximation, since in practice the last $N_w - 1$ samples of h_l^{ISI} are generally very close to zero.

In matrix-vector notation,

$$\begin{aligned} H_i^{signal} &= \mathbf{q}_i^H \mathbf{G} \mathbf{H} \mathbf{w} \\ H_i^{ISI} &= \mathbf{q}_i^H \mathbf{D} \mathbf{H} \mathbf{w} \\ W_i &= \mathbf{q}_i^H \mathbf{F} \mathbf{w} \end{aligned} \quad (3)$$

where

$$\begin{aligned} \mathbf{w} &= [w_0 \ w_1 \ \dots \ w_{N_w-1}]^T \\ \mathbf{H} &= \begin{bmatrix} h_0 & h_{-1} & \dots & h_{-(N_w-1)} \\ h_1 & h_0 & \dots & h_{-(N_w-2)} \\ \vdots & \vdots & \ddots & \vdots \\ h_{N-1} & h_{N-2} & \dots & h_{-(N-N_w)} \end{bmatrix} \\ \mathbf{G} &= \text{diag}[g_0 \ g_1 \ \dots \ g_{N-1}]^T \\ \mathbf{D} &= \mathbf{I} - \mathbf{G} \\ \mathbf{q}_i &= [1 \ e^{j2\pi i/N} \ \dots \ e^{j2\pi i(N-1)/N}]^T \end{aligned}$$

So the SNR in subchannels can be written as

$$\text{SNR}_i = \frac{S_{x,i} |\mathbf{q}_i^H \mathbf{G} \mathbf{H} \mathbf{w}|^2}{S_{n,i} |\mathbf{q}_i^H \mathbf{F} \mathbf{w}|^2 + S_{x,i} |\mathbf{q}_i^H \mathbf{D} \mathbf{H} \mathbf{w}|^2} \quad (4)$$

where $\mathbf{q}_i^H \mathbf{F} \mathbf{w}$ is the i^{th} N -point FFT coefficient of \mathbf{w} .

Using the SNR definition in (4) the channel capacity of the DMT transceiver can be written as

$$b_{\text{DMT}} = \sum_{i=0}^{N-1} \log_2 \left(1 + \frac{\text{SNR}_i}{\gamma} \right) \quad (5)$$

where γ is the desired system margin [3]. By maximizing this function with respect to \mathbf{w} , we can obtain the maximum channel capacity (MCC) TEQ [2].

The objective function in (5) is a nonlinear function of \mathbf{w} , which requires nonlinear optimization methods. The min-ISI method simplifies the optimization problem by minimizing the total distortion power instead of maximizing channel capacity. Since a power term is always nonnegative, minimizing the distortion power in each subchannel (the denominator of SNR_i in (4)) is equivalent to minimizing the sum of the distortion power of all subchannels, which can be written as [2]

$$\mathbf{w}^T \mathbf{H}^T \mathbf{D}^T \sum_{i \in \mathcal{S}} \left(\mathbf{q}_i |H_i|^2 \frac{S_{x,i}}{S_{n,i}} \mathbf{q}_i^H \right) \mathbf{D} \mathbf{H} \mathbf{w} = \mathbf{w}^T \mathbf{A} \mathbf{w} \quad (6)$$

To prevent minimization of the signal power, we constrain the signal path impulse response energy to one:

$$\|\mathbf{h}^{signal}\|^2 = \mathbf{w}^T \mathbf{H}^T \mathbf{G}^T \mathbf{G} \mathbf{H} \mathbf{w} = \mathbf{w}^T \mathbf{B} \mathbf{w} = 1 \quad (7)$$

Hence, the output signal power is equal to the input signal power. Finally, the optimization problem for minimum ISI becomes

$$\min_{\mathbf{w}} \mathbf{w}^T \mathbf{A} \mathbf{w} \quad \text{s.t.} \quad \mathbf{w}^T \mathbf{B} \mathbf{w} = 1 \quad (8)$$

Solving (8) with respect to \mathbf{w} gives results very close to that of maximizing (5). We rewrite (6) and (7) as

$$\mathbf{A} = (\mathbf{Q} \mathbf{D} \mathbf{H})^H \text{diag}(\mathbf{S}) (\mathbf{Q} \mathbf{D} \mathbf{H}) \quad (9)$$

$$\mathbf{B} = (\mathbf{G} \mathbf{H})^T (\mathbf{G} \mathbf{H}) \quad (10)$$

where $\mathbf{Q} = [\mathbf{q}_0 \ \mathbf{q}_1 \ \dots \ \mathbf{q}_{N-1}]^T$ and \mathbf{S} is a vector of length N containing the subchannel weighting such that $S_i = |H_i|^2 S_{x,i}/S_{n,i}$.

3. Recursive Min-ISI Method

In (10), we can view both \mathbf{A} and \mathbf{B} as the multiplication of an $N_w \times N$ matrix with an $N \times N_w$ matrix, where $N_w \ll N$. Thus it makes sense precompute \mathbf{A} and \mathbf{B} for the optimization problem, since both matrices have dimensions of only $N_w \times N_w$. The small size of \mathbf{A} and \mathbf{B} allows the solution of the optimization problem to be found rapidly by the inverse power method or some other means. However, naively computing \mathbf{A} and \mathbf{B} is very expensive for a real-time implementation. Hence this section will take advantage of the structure in the argument matrices, especially the Toeplitz structure of \mathbf{H} , in developing an efficient algorithm to compute \mathbf{A} and \mathbf{B} .

3.1. Fast calculation of matrix \mathbf{B}

A fast algorithm to calculate \mathbf{B} is straightforward. The n^{th} column of the convolution matrix \mathbf{H} can be written as

$$\mathbf{H}_n = [h_{0-n} \ h_{1-n} \ \dots \ h_{N-1-n}]^T$$

Left multiplication by the windowing matrix \mathbf{G} gives

$$\mathbf{GH}_n = [\mathbf{0} \ h_{\Delta-n} \ \dots \ h_{\Delta+\nu-n} \ \mathbf{0}]^T$$

where $\mathbf{0}$ denotes a row of zeros. Then the elements of \mathbf{B} can be given as

$$B_{m,n} = (\mathbf{GH}_m)^T (\mathbf{GH}_n) = \sum_{k=\Delta}^{\Delta+\nu} h_{k-m} h_{k-n} \quad (11)$$

From (11) we note that

$$B_{m+1,n+1} = \sum_{k=\Delta-1}^{\Delta+\nu-1} h_{k-m} h_{k-n}$$

which arrives us at the following recursive formula on the diagonals of \mathbf{B} :

$$\begin{aligned} B_{m+1,n+1} &= B_{m,n} \\ &\quad - h_{\Delta+\nu-m} h_{\Delta+\nu-n} \\ &\quad + h_{\Delta-1-m} h_{\Delta-1-n} \end{aligned} \quad (12)$$

\mathbf{B} can then be computed as follows. First, use (11) to directly calculate the first column of \mathbf{B} . This takes $N_w(\nu+1)$ multiply-accumulate (MAC) operations. Then use the recursion in (12) to fill out

the lower triangular part of \mathbf{B} . This will take only $N_w(N_w-1)$ MACs to complete. Since \mathbf{B} is symmetric, we are done calculating \mathbf{B} using only $N_w(\nu+N_w)$ MACs.

3.2. Fast calculation of matrix \mathbf{A}

We will proceed in a similar fashion to produce an algorithm to compute \mathbf{A} . We start by writing out the n^{th} column of \mathbf{DH} :

$$\mathbf{DH}_n = \begin{bmatrix} h_{0-n} & \dots & h_{\Delta-1-n} & \mathbf{0} \\ h_{\Delta+\nu+1-n} & \dots & h_{N-1-n} & \end{bmatrix}^T$$

Since left-multiplying by \mathbf{Q} produces the N -point FFT for each column of \mathbf{DH}_n , then it follows that

$$(\mathbf{QDH})_{m,n} = \left(\sum_{k=0}^{\Delta-1} + \sum_{k=\Delta+\nu+1}^{N-1} \right) h_{k-n} e^{-j2\pi mk/N}$$

By using the fact that

$$A_{m,n} = \sum_{p=0}^{N-1} (\mathbf{QDH})_{m,p}^H S_p (\mathbf{QDH})_{p,n}$$

we have the following formula for the elements of \mathbf{A} :

$$A_{m,n} = \sum_{k=0}^{\Delta-1} \sum_{l=0}^{\Delta-1} h_{k-m} h_{l-n} s_{k-l} \quad (13)$$

$$+ \sum_{k=0}^{\Delta-1} \sum_{l=\Delta+\nu+1}^{N-1} h_{k-m} h_{l-n} s_{k-l} \quad (14)$$

$$+ \sum_{k=\Delta+\nu+1}^{N-1} \sum_{l=0}^{\Delta-1} h_{k-m} h_{l-n} s_{k-l} \quad (15)$$

$$+ \sum_{k=\Delta+\nu+1}^{N-1} \sum_{l=\Delta+\nu+1}^{N-1} h_{k-m} h_{l-n} s_{k-l} \quad (16)$$

where we define s_{k-l} as

$$s_{k-l} = \sum_{p=0}^{N-1} S_p e^{j2\pi(k-l)p/N} \quad (17)$$

Thus s_{k-l} can be viewed as an inverse FFT component of the vector \mathbf{S} . Since \mathbf{S} is both real and symmetric about the Nyquist frequency, s_{k-l} is always real, and furthermore $s_{l-k} = s_{k-l}$.

The double sums in (13)–(16) allow us to decompose \mathbf{A} into the sum of four matrices, each of the form

$$C_{m,n} = \sum_{k=a}^b \sum_{l=c}^d h_{k-m} h_{l-n} s_{k-l} \quad (18)$$

As before, we notice that

$$C_{m+1,n+1} = \sum_{k=a-1}^{b-1} \sum_{l=c-1}^{d-1} h_{k-m} h_{l-n} s_{k-l}$$

From this realization, we can prove the following recursive formula for the diagonals of \mathbf{C} :

$$\begin{aligned} C_{m+1,n+1} &= C_{m,n} \\ &\quad - h_{b-m} h_{d-n} s_{b-d} \\ &\quad - h_{b-m} f(c, d, b, n) \\ &\quad - h_{d-n} f(a, b, d, m) \\ &\quad + h_{a-1-m} h_{c-1-n} s_{a-c} \\ &\quad + h_{a-1-m} f(c, d, a-1, n) \\ &\quad + h_{c-1-n} f(a, b, c-1, m) \end{aligned} \quad (19)$$

where we define the function f as

$$f(\alpha, \beta, \gamma, \delta) = \sum_{k=\alpha}^{\beta-1} h_{k-\delta} s_{k-\gamma} \quad (20)$$

3.3. Computational complexity analysis

Each occurrence of the function f in (19) has the first three parameters constant, and the last parameter either m or n . Since there are only $N_w - 1$ possible values of m or n for the recursion, so there are only $4(N_w - 1)$ different values of f . Half of these values will require $b - a$ MACs each, and the other half will require $d - c$ MACs each, so we will need $2(N_w - 1)(b - a + d - c)$ MACs to compute all possible values of f . In the special case that $a = c$ and $b = d$ (i.e. in the double sums (13) and (16)), we only have $2(N_w - 1)$ different function evaluations, thus needing only $2(N_w - 1)(b - a)$ MACs to compute all values of f . Using the recursive formula to fill in the lower triangular part of \mathbf{C} will take at most $4N_w(N_w - 1)$ extra MACs to complete.

An immediate savings in computation can be realized by noticing that matrices formed by the double sums in (14) and (15) are actually transposes of each other. As a result, their corresponding recursive formulas are the same except for the fact that m and n are swapped. This allows both matrices to use the same set of values of f for their recursive formulas.

Of course, we still need the first column of \mathbf{C} in order to use the recursion in (19). We cannot directly use (18) to compute the first column—such a method would be far too inefficient. A better solution is to use the matrix equation for \mathbf{A} in (9), reformulated here to express the desired order of multiplication:

$$\mathbf{A}_0 = (\mathbf{D}\mathbf{H})^T \mathbf{Q}^H \text{diag}(\mathbf{S}) \mathbf{Q} \mathbf{D} \mathbf{H}_0 \quad (21)$$

where \mathbf{A}_0 denotes the first column of \mathbf{A} . By using the above formula, we can compute \mathbf{A}_0 using $2 N$ -point FFTs and $N + N_w(N - \nu - 1)$ MACs. Since many DMT systems have hardware FFTs, a possible implementation is to offload the FFT operations to the hardware while a DSP performs the other calculations.

Notice that approach directly calculates the first column of \mathbf{A} instead of the first columns of the individual double sums. Thus, we simply add the four recursive formulas from the double sums of \mathbf{A} together and use the resulting formula to fill the lower triangular part of \mathbf{A} . The total amount of computations needed to compute both \mathbf{A} and \mathbf{B} then becomes $2 N$ -point FFTs and $4(N_w - 1)(N + 4N_w - \nu - 2) + N_w(N + N_w - 1) + N$ MACs. In practice, N is much larger than both ν and N_w , so the number of MACs required approximates to about $5N_w N$.

4. Row-Rotation Min-ISI Method

We can reduce the computational complexity of our method by noticing that if $\Delta = 0$, then the double sums in (13), (14), and (15) vanish to zero. One possible way to effectively force Δ to be zero is to introduce the “row rotation” matrix \mathbf{R} , defined by the equation

$$\mathbf{R} = [\mathbf{e}_{N-\Delta} \quad \dots \quad \mathbf{e}_{N-1} \quad \mathbf{e}_0 \quad \dots \quad \mathbf{e}_{N-\Delta-1}] \quad (22)$$

where $\mathbf{e}_0, \dots, \mathbf{e}_{N-1}$ are the standard basis vectors in \mathcal{R}^n . \mathbf{R} has the following properties:

$$\mathbf{Q}\mathbf{R} = \text{diag}(\mathbf{q}_\Delta)^H \mathbf{Q} \quad (23)$$

$$\mathbf{R} \text{diag}(\mathbf{d}) = \text{diag}(\mathbf{R}\mathbf{d}) \mathbf{R} \quad (24)$$

where \mathbf{d} can be any N -dimensional vector, but in this case represents the diagonal of the windowing matrix \mathbf{D} . Using these properties of \mathbf{R} , we have following equivalent form of \mathbf{A} :

$$\mathbf{A} = (\mathbf{Q} \text{diag}(\mathbf{R}\mathbf{d}) \mathbf{R}\mathbf{H})^H \text{diag}(\mathbf{S}) (\mathbf{Q} \text{diag}(\mathbf{R}\mathbf{d}) \mathbf{R}\mathbf{H}) \quad (25)$$

Here, (25) has the same structure as the original equation for \mathbf{A} in (9), except for the possibility that $\mathbf{R}\mathbf{H}$ is not Toeplitz. So as it stands, we cannot use our recursive methods here because they depend on the Toeplitz nature of \mathbf{H} . However, $\mathbf{R}\mathbf{H}$ is guaranteed to be Toeplitz if the last $N_w - 1$ samples of the channel impulse response are zero. In practice, the last $N_w - 1$ samples are indeed extremely close to zero, and so we lose almost no accuracy by modeling the channel with $N - N_w + 1$ coefficients padded with $N_w - 1$ zeros. Such an approach is more natural in some respects because now the convolution of the channel and equalizer is fully represented by the matrix multiplication $\mathbf{H}\mathbf{w}$.

With this slight change to the channel impulse response, we can write the following equation for \mathbf{A}

$$A_{m,n} = \sum_{k=\nu+1}^{N-1} \sum_{l=\nu+1}^{N-1} h_{R,k-m} h_{R,l-n} s_{k-l} \quad (26)$$

where $h_{R,m-n}$ is the element at the m^{th} row and n^{th} column of the *Toeplitz* matrix \mathbf{RH} . This formulation reduces the complexity of computing \mathbf{A} and \mathbf{B} to 2 FFTs and $2(N_w - 1)(N + 2N_w - \nu - 2) + N_w(N + N_w - 1) + N$ MACs, or approximately $3N_w N$ MACs.

5. No-Weighting Min-ISI Method

Suppose that we ignore the subchannel weighting by assuming that each subchannel has equal importance in the cost function. Then the matrix $\text{diag}(\mathbf{S})$ becomes a multiple of the identity matrix, and the orthogonality of the FFT matrix \mathbf{Q} gives

$$\mathbf{A} = M(\mathbf{DH})^T(\mathbf{DH}) \quad (27)$$

where M is some positive scalar. This assumption essentially reduces the min-ISI method to the maximum SSNR method [5]. We can use the same methods used to derive (11) and (12) to devise a fast algorithm to calculate \mathbf{A} . Since this simplified method does not require the use of an FFT, we are not limited by the size of the vectors in the convolution matrix \mathbf{H} . Hence we can use a full convolution matrix with $N + N_w - 1$ rows if desired, provided that we also enlarge the diagonal matrices \mathbf{G} and \mathbf{D} accordingly.

This simplified method reduces the calculation of \mathbf{A} and \mathbf{B} to $N_w N + 5N_w(N_w - 1)$ MACs—without the need for any FFT operations. While equally weighing the subchannels is indeed a major simplification, the maximum SSNR method performs fairly well in practice. Therefore, this simplified method is a reasonable approach if computational resources are at a premium.

6. Results

Table 1 summarizes the computational complexity of the TEQ design methods which require computation of \mathbf{A} and \mathbf{B} . The complexities for the original minimum ISI method and maximum SSNR method was derived from the computation of the formulas in (9), (10), and (27) where the optimal order of matrix multiplication is specified by the parentheses of the formulas. Computing \mathbf{QDH} efficiently involves direct computation of the first column using an FFT, and the application of sliding window methods to compute the rest of the columns. We remark that to keep the comparisons fair,

Method	Complexity (excluding FFTs)
original min-ISI	$\frac{1}{2}(N + \nu)N_w(N_w + 1) + 5N(N_w - 1) + NN_w$
recursive min-ISI	$4(N_w - 1)(N + 4N_w - \nu - 2) + N_w(N + N_w - 1) + N$
row-rotation min-ISI	$2(N_w - 1)(N + 2N_w - \nu - 2) + N_w(N + N_w - 1) + N$
maximum SSNR	$\frac{1}{2}NN_w(N_w + 1)$
no-weighting min-ISI	$NN_w + 5N_w(N_w + 1)$

Table 1. Comparison of the computational complexities of two previous methods (original min-ISI and maximum SSNR) with the three proposed methods.

all obvious optimizations, such as complex conjugate mirroring and the real, symmetric nature of \mathbf{A} and \mathbf{B} were taken into consideration when computing these complexities.

Table 2 shows the performance and computational complexity of the TEQ design methods obtained from simulations. All of the methods achieve close to 100% channel capacity, with a slight gap between methods that include subchannel weighting and those that do not. The methods proposed in this paper drastically reduce the complexity of calculating \mathbf{A} and \mathbf{B} . In addition, we expect the complexity gap to increase as N_w increases, and the performance gap to increase as the spectral shaping in the channel becomes more severe.

The results in Table 2 also show that maximizing the SSNR does not necessarily maximize channel capacity. Although the maximum SSNR method [5] gives higher SSNR, the achievable channel capacity is lower than the min-ISI base methods. Fig. 1 shows a screen shot of a Matlab DMTTEQ toolbox developed by Arslan, Lu, and Evans, which is available at

<http://signal.ece.utexas.edu/~arslan/dmtteq/>

References

- [1] G. Arslan, B. L. Evans, and S. Kiaei. Optimum channel shortening for discrete multitone transceivers. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing*, Istanbul, Turkey, June 2000.
- [2] G. Arslan, B. L. Evans, and S. Kiaei. Equalization for discrete multitone transceivers to maximize channel capacity. *IEEE Trans. on Signal Processing*, submitted.

Method	Capacity percentage	SSNR (dB)	MACs
original min-ISI	99.6%	37.8	132896
recursive min-ISI	99.5%	37.9	44432
row-rotation min-ISI	99.5%	37.5	25872
maximum SSNR	97.9%	58.9	78836
no-weighting min-ISI	97.8%	55.4	10064

Table 2. Comparison of two previous methods (original min-ISI and maximum SSNR) with the three proposed methods for CSA loop 1, FFT size of 512, cyclic prefix length of 32, a TEQ of 17 taps, and a system margin of 6 dB.

- [3] J. M. Cioffi. *A Multicarrier Primer*. Amati Communication Corporation and Stanford University, T1E1.4/91-157, Nov. 1991.
- [4] B. Lu, L. D. Clark, G. Arslan, and B. L. Evans. Divide-and-conquer and matrix pencil methods for discrete multitone equalization. *IEEE Trans. on Signal Processing*, submitted.
- [5] P. J. W. Melsa, R. C. Younce, and C. E. Rohrs. Impulse response shortening for discrete multitone transceivers. *IEEE Trans. on Comm.*, 44(12):1662–1672, Dec. 1996.

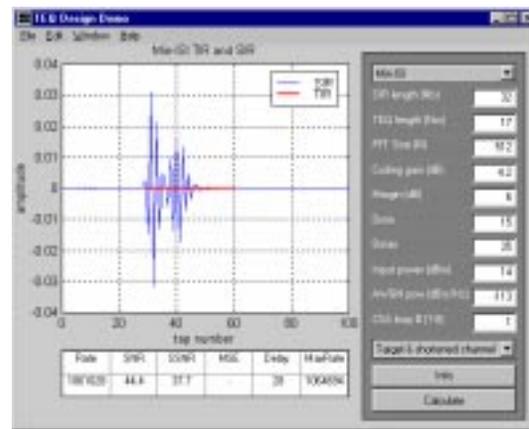


Figure 1. A snapshot of the MATLAB DMTEQ toolbox, which was used for all simulations.